(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 851 358 B1

(12) EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:
05.03.2003 Bulletin 2003/10

(51) Int Cl.7: G06F 12/14, G06F 1/00

(21) Application number: 97310176.9

(22) Date of filing: 16.12.1997

(54) Processing system security

Datenverarbeitungssystemsicherheit

Sécurité pour système de traitement de données

(84) Designated Contracting States:
DE FR GB IT NL SE

(30) Priority: 31.12.1996 US 777256

(43) Date of publication of application:
01.07.1998 Bulletin 1998/27

(73) Proprietor: Sun Microsystems, Inc.
Palo Alto, California 94303-4900 (US)

(72) Inventor: Garnett, Paul Jeffrey
Merseyside, WA12 9PW (GB)

(74) Representative: Harris, Ian Richard et al
D. Young & Co.,
21 New Fetter Lane
London EC4A 1DA (GB)

(56) References cited:
EP-A- 0 660 215
DE-A- 2 733 531
US-A- 5 535 409

Description

[0001] This invention relates to computer system security, and in particular to the protection of a critical resource within a processing system.

[0002] A computer system may be stopped from working by a software bug, or by a malicious act interfering with a critical system resource. The system resource can be a hardware resource, or can be software or data stored or otherwise retained within the computing system.

[0003] DE 2733531 A, US 5535409A and EP 660215 A disclose prior art arrangements for controlling access to memory resources.

[0004] An object of the invention is to improve the security of a system by preventing, or making more difficult, access to a critical resource.

[0005] In accordance with a first aspect of the invention, there is provided a method of controlling access to a system resource in a processing system including a processing engine, said system resource and a bus, said method comprising steps of: in a first mode programming reprogrammable logic located between said bus and said resource to permit access by said processing engine via said bus to said resource; and in a second mode programming said reprogrammable logic to at least restrict access to said resource via said bus.

[0006] By reprogramming the reprogrammable logic to restrict access to the critical resource, accidental or deliberate corruption of the critical resource can be prevented or at least made more difficult, thus improving overall system security and integrity.

[0007] The resource can be a critical hardware component, although it will typically be an area of storage containing or identifying critical operational software or critical operational parameters of the system. The programming of the reprogrammable logic in the second mode can thus be used to block access selectively or totally to the storage area.

[0008] The storage area can comprise test software and/or test parameters relating to the processing system.

[0009] Preferably, the reprogrammable logic comprises a field programmable gate array. First programming information and second programming information can both be stored in memory in the processing system. Alternatively the first programming information is supplied externally to the processing system when required, and is not otherwise stored in the processing system. The latter alternative provides higher security and may be particularly useful for test purposes, whereby a test engineer could supply the first programming information in a test mode of operation.

[0010] In accordance with a further aspect of the invention, there is provided a processing system (for example a computer system) comprising a processing engine (for example a microprocessor or microcontroller), a system resource, a bus and reprogrammable logic located between said bus and said resource, said reprogrammable logic being programmable in a first mode to permit access by said processing engine via said bus to said resource and being programmable in a second mode to at least restrict access to said resource via said bus.

[0011] Particular embodiments of the invention will be described hereinafter with reference to the accompanying drawings in which like reference signs relate to like elements and in which:

- Figure 1 is a schematic overview of a computing system;
- Figure 2 is a schematic overview of one implementation of the present invention;
- Figures 3-6 relate to further examples of implementations of the present invention;
- Figure 7 is a schematic diagram illustrating the operation of the invention.

[0012] Figure 1 is a schematic overview of a computing system 10. The computing system comprises a processor 20 connected via a bus 30 to a number of resources. A display adaptor 22 enables a display 24 to be connected to the bus 30. Similarly, a keyboard adaptor 26 allows a keyboard 28 to be connected to the bus 30. A first memory M1 32 is connected directly to the bus 30. A further memory M2 38 containing a critical software resource 40 is connected to the bus 30 via reprogrammable logic, for example a reprogrammable gate array 36.

[0013] A communications adaptor 42 enables an external communications line 44 to be connected to the computer bus 30.

[0014] It will be appreciated that Figure 1 is merely a schematic overview of a computing system, and that an alternative computing system could have a different structure from that shown in Figure 1. In the example of Figure 1, and also in the examples set out hereinafter, the critical resource is assumed to be data, or software, stored in the memory M2 38. However, in other embodiments of the invention the critical resource could be a hardware component which is, for example, reconfigurable. An example of such a component could be a clock signal generator having a selectable clock rate. Alternatively, the hardware component could be a component of a system which is used only during certain modes of operation of the system. It may be desirable to prevent access to the hardware component by the user of the apparatus, permitting access only during a test or repair mode under the control of a service engineer.

[0015] In a preferred embodiment of the invention, the reprogrammable logic 36 is a reprogrammable gate array, such as a field programmable gate array (FPGA). In the following embodiments reference will be made to an FPGA 36, although other implementations of the device for restricting access to the resource may be pro-

vided.

[0016] Figure 7 illustrates an initial stage of operation of an embodiment of the invention such as that described in Figure 1. In particular, on initially booting the computing system, initialisation code 50 is operable to carry out the initialisation of the system. At a point in the initialisation program, reference C1 is made to first code stored at locations 52 in the memory M1 32. This information is loaded from the memory 32 (which could, for example, be a read only memory) into the FPGA 36 to initially program the FPGA 36. The FPGA 36 is then responsive to signals provided over the computer bus 30 to provide a first mode for enabling access by programming code P to the critical resource 40 in the memory M2 to perform the appropriate initialisation functions. After execution of the code P, a second reference C2 to information stored at 54 in the memory 32 is made. This second information 54 is then loaded into the FPGA 36 overwriting the original programming of the FPGA 36. This reprogramming of the FPGA 36 can then prevent, or restrict further access to the critical resource 40 in a second mode.

[0017] Any one of many different models of commercially available reprogrammable logic can be employed as the FPGA 36 in an embodiment of the invention. The programming of the FPGA 36 will depend on the particular reprogrammable FPGA used and should be in accordance with the technical design specification for the FPGA concerned, as will be apparent to one skilled in the art.

[0018] Figure 2 is a schematic representation of one configuration of an FPGA 36 and a memory 38, the FPGA receiving an N bit bus, and being connected to the memory 38 by an M bit bus.

[0019] Figure 3 represents one specific embodiment of the invention with an FPGA receiving a read/write line r/w and a chip select line at cs and supplying the chip select and read/write lines to the memory 38. In this embodiment, the FPGA can be initially programmed to pass read and write signals, as well as the chip select signals to the memory 38. With the second programming (reprogramming) the FPGA can be arranged to pass only read signals, thus preventing writing to the memory 38. Alternatively, both read and write signals to the memory 38 could be prevented with the second programming. In either case system resource protection is provided either by preventing overwriting by the user or by completely blocking access to the memory 38.

[0020] Figures 4 and 5 indicate that the FPGA 36 can provide a translation between different sized buses. Thus, in Figure 4, a narrow bus B1 accesses the FPGA 36 but the FPGA 36 decodes the signals on the input bus B1 to provide individual decoded lines B2 for accessing the memory 38.

[0021] In Figure 5, a wide bus B3 accesses the FPGA 36, which only passes selected bits B4 of the address to the memory 38. In an embodiment of the invention, the FPGA provides a different translation in the first and second programming modes.

[0022] Figure 6 illustrates an arrangement where a bus B5 is provided at the input side of the FPGA 36. At the output side of FPGA, address lines represented at B6 and further address lines represented at B6' are provided. In an embodiment of the invention the address lines B6 provide address signals with both the first and second programming of the PFGA, whereas the address signals on lines B6' are either blocked or altered with either the first or second programming of the FPGA, whereby with the first programming, access to test software 44 within the memory 38 can be permitted, whereas with the second programming, access to the test programming is prevented and access is instead permitted to user software 42. It would then be possible, during initial design and testing, or during subsequent use for diagnostic purposes, for a test engineer to reprogram the FPGA with the original, first programming, and thereby permit access once more to the test software 44.

[0023] As illustrated schematically in Figure 1, the information 52, 54 is provided in the memory area 34 of memory 32. However, it may be that the information relating to the first programming of the FPGA may not be retained in the memory of the computer system 10 at all, but could instead be provided externally by a test engineer when testing of the system is required. In this way, it would not be possible for the user to seek out and possibly find the information required for programming the FPGA.

[0024] To provide additional security, it would be possible to encrypt the data stored in the areas 52/54 to make it more difficult for a user to find the information necessary for programming the FPGA 36.

[0025] An embodiment of the invention can provide security of operation in that accidental access to critical resources can be prevented during normal use. Also, an embodiment of the invention can make it significantly more difficult for the user to access the critical resources.

[0026] Although particular embodiments of the inventions have been described, it will be appreciated that the invention is not limited thereto, and many modifications and/or additions may be made within the scope of the invention as defined in the claims.

[0027] Also, by way of further example, although in the embodiment described herein, the processing system is shown as a computer system, the processing system could be any apparatus or system having a computer-based, microprocessor-based or microcontroller-based control system.

Claims

1. A method of controlling access to a system resource (40) in a processing system (10) including a processing engine (20), said system resource and

a bus (30), said method comprising steps of:

in a first mode programming reprogrammable logic (36) located between said bus and said resource to permit access by said processing engine via said bus to said resource; and
in a second mode programming said reprogrammable logic to at least restrict access to said resource via said bus.

2. A method according to Claim 1, wherein said resource is a critical area of storage, said second programming of said reprogrammable logic blocking access to said storage area.

3. A method according to Claim 2, wherein said storage area comprises critical operational parameters relating to said processing system.

4. A method according to Claim 2, wherein said storage area identifies or comprises critical operational software.

5. A method according to Claim 2, wherein said storage area comprises test software and/or test parameters relating to said processing system.

6. A method according to any preceding Claim, wherein said reprogrammable logic comprises a reprogrammable field programmable gate array.

7. A method according to Claim 6, wherein first programming information and second programming information are stored in memory in said processing system.

8. A method according to any preceding Claim, wherein said first programming information is supplied externally to said processing system when required, and is not otherwise stored in said processing system.

9. A processing system (10) comprising a processing engine (20), system resource (40), a bus (30) and reprogrammable logic (36) located between said bus and said resource, said reprogrammable logic being programmable in a first mode to permit access by said processing engine via said bus to said resource and being programmable in a second mode to at least restrict access to said resource via said bus.

10. A system according to Claim 9, wherein said critical resource is an area of storage (38), said reprogrammable logic, when programmed in said second mode, blocking access to said storage area.

11. A system according to Claim 10, wherein said storage area comprises critical operational parameters relating to said processing system.

12. A system according to Claim 10, wherein said storage area identifies or comprises critical operational software.

13. A system according to Claim 10, wherein said storage area comprises test software and/or test parameters relating to said processing system.

14. A system according to any one of Claims 9 to 13, wherein said reprogrammable logic comprises a reprogrammable field programmable gate array.

15. A system according to Claim 14, wherein first programming information and second programming information are stored in memory in said processing system.

16. A system according to any one of Claims 9 to 15, wherein said first programming information is supplied externally to said processing system when required, and is not otherwise stored in said processing system.

Patentansprüche

1. Verfahren zur Kontrolle des Zugriffs auf eine Systemresource (40) in einem Verarbeitungssystem (10), welches eine Verarbeitungsmaschine (20), die Systemresource und einen Bus (30) umfaßt, wobei das Verfahren die Schritte aufweist:

in einer ersten Betriebsart Programmieren einer wiederprogrammierbaren Logik (36), die zwischen dem Bus und der Resource angeordnet ist, um einen Zugriff durch die Verarbeitungsmaschine über den Bus auf die Resource zu ermöglichen, und
in einer zweiten Betriebsart Programmieren der wiederprogrammierbaren Logik, um den Zugriff über den Bus auf die Resource zumindest einzuschränken.

2. Verfahren nach Anspruch 1, wobei die Resource ein kritischer Bereich eines Speichers ist, und wobei die zweite Programmierung der wiederprogrammierbaren Logik den Zugriff zu diesem Speicherbereich blockiert.

3. Verfahren nach Anspruch 2, wobei der Speicherbereich kritische Betriebsparameter aufweist, die sich auf das Verarbeitungssystem beziehen.

4. Verfahren nach Anspruch 2, wobei der Speicherbereich kritische Betriebssoftware identifiziert oder

aufweist.

5. Verfahren nach Anspruch 2, wobei der Speicherbereich Testsoftware und/oder Testparameter aufweist, die sich auf das Verarbeitungssystem beziehen.

6. Verfahren nach irgendeinem der vorstehenden Ansprüche, wobei die wiederprogrammierbare Logik ein wiederprogrammierbares, feldprogrammierbares Gatearray aufweist.

7. Verfahren nach Anspruch 6, wobei die erste Programmierinformation und die zweite Programmierinformation in dem Speicher in dem Verarbeitungssystem gespeichert sind.

8. Verfahren nach irgendeinem der vorstehenden Ansprüche, wobei die erste Programmierinformation dem Verarbeitungssystem extern zugeführt wird, wenn dies erforderlich ist, und im übrigen nicht in dem Verarbeitungssystem gespeichert wird.

9. Verarbeitungssystem (10), mit einer Verarbeitungsmaschine (20), einer Systemresource (40), einem Bus (30) und einer wiederprogrammierbaren Logik 36, die zwischen dem Bus und der Resource angeordnet ist, wobei die wiederprogrammierbare Logik so programmierbar ist, daß sie in einer ersten Betriebsart den Zugriff durch die Verarbeitungsmaschine über den Bus auf die Resource ermöglicht und in einer zweiten Betriebsart so programmierbar ist, daß sie den Zugriff auf die Resource über den Bus zumindest einschränkt.

10. System nach Anspruch 9, wobei die kritische Resource ein Bereich eines Speichers (38) ist, wobei die wiederprogrammierbare Logik, wenn sie in der zweiten Betriebsart programmiert ist, den Zugriff auf den Speicherbereich blockiert.

11. System nach Anspruch 10, wobei der Speicherbereich kritische Betriebsparameter aufweist, die sich auf das Verarbeitungssystem beziehen.

12. System nach Anspruch 10, wobei der Speicherbereich kritische Betriebssoftware identifiziert oder aufweist.

13. System nach Anspruch 10, wobei der Speicherbereich Testsoftware und/oder Testparameter aufweist, die sich auf das Verarbeitungssystem beziehen.

14. System nach einem der Ansprüche 9 bis 13, wobei die wiederprogrammierbare Logik ein wiederprogrammierbares, feldprogrammierbares Gatearray aufweist.

15. System nach Anspruch 14, wobei erste Programmierinformationen und zweite Programmierinformationen in dem Speicher in dem Verarbeitungssystem gespeichert sind.

16. System nach einem der Ansprüche 9 bis 15, wobei die erste Programmierinformation dem Verarbeitungssystem extern zugeführt wird, wenn dies erforderlich ist, und im übrigen nicht in dem Verarbeitungssystem gespeichert wird.

Revendications

1. Procédé pour commander l'accès à une ressource de système (40) dans un système de traitement (10) incluant un moteur de traitement (20), la ressource de système et un bus (30), ce procédé comprenant les étapes suivantes :

   dans un premier mode, on programme une logique reprogrammable (36) placée entre le bus et la ressource pour permettre l'accès à la ressource par le moteur de traitement, par l'intermédiaire du bus; et
   dans un second mode, on programme la logique reprogrammable pour au moins restreindre l'accès à la ressource par l'intermédiaire du bus.

2. Procédé selon la revendication 1, dans lequel la ressource est une zone de stockage critique, la seconde programmation de la logique reprogrammable bloquant l'accès à cette zone de stockage.

3. Procédé selon la revendication 2, dans lequel la zone de stockage contient des paramètres opérationnels critiques concernant le système de traitement.

4. Procédé selon la revendication 2, dans lequel la zone de stockage identifie ou contient un logiciel opérationnel critique.

5. Procédé selon la revendication 2, dans lequel la zone de stockage contient un logiciel de test et/ou des paramètres de test concernant le système de traitement.

6. Procédé selon l'une quelconque des revendications précédentes, dans lequel la logique reprogrammable comprend un réseau de portes programmable sur les lieux d'utilisation et reprogrammable.

7. Procédé selon la revendication 6, dans lequel la première information de programmation et la seconde information de programmation sont stockées en mémoire dans le système de traitement.

8. Procédé selon l'une quelconque des revendications précédentes, dans lequel la première information de programmation est fournie à partir de l'extérieur au système de traitement lorsque c'est nécessaire, et elle n'est pas par ailleurs stockée dans le système de traitement.

9. Système de traitement (10) comprenant un moteur de traitement (20), une ressource de système (40), un bus (30) et une logique reprogrammable (36) placée entre le bus et la ressource, cette logique reprogrammable étant programmable dans un premier mode pour permettre l'accès à la ressource par le moteur de traitement, par l'intermédiaire du bus, et étant programmable dans un second mode pour au moins restreindre l'accès à la ressource par l'intermédiaire du bus.

10. Système selon la revendication 9, dans lequel la ressource critique est une zone de stockage (38), et lorsque la logique reprogrammable est programmée dans le second mode, elle bloque l'accès à cette zone de stockage.

11. Système selon la revendication 10, dans lequel la zone de stockage contient des paramètres opérationnels critiques concernant le système de traitement.

12. Système selon la revendication 10, dans lequel la zone de stockage identifie ou contient un logiciel opérationnel critique.

13. Système selon la revendication 10, dans lequel la zone de stockage contient un logiciel de test et/ou des paramètres de test concernant le système de traitement.

14. Système selon l'une quelconque des revendications 9 à 13, dans lequel la logique reprogrammable comprend un réseau de portes programmable sur les lieux d'utilisation et reprogrammable.

15. Système selon la revendication 14, dans lequel une première information de programmation et une seconde information de programmation sont stockées en mémoire dans le système de traitement.

16. Système selon l'une quelconque des revendications 9 à 15, dans lequel la première information de programmation est fournie au système de traitement à partir de l'extérieur lorsque c'est nécessaire, et elle n'est pas par ailleurs stockée dans le système de traitement.

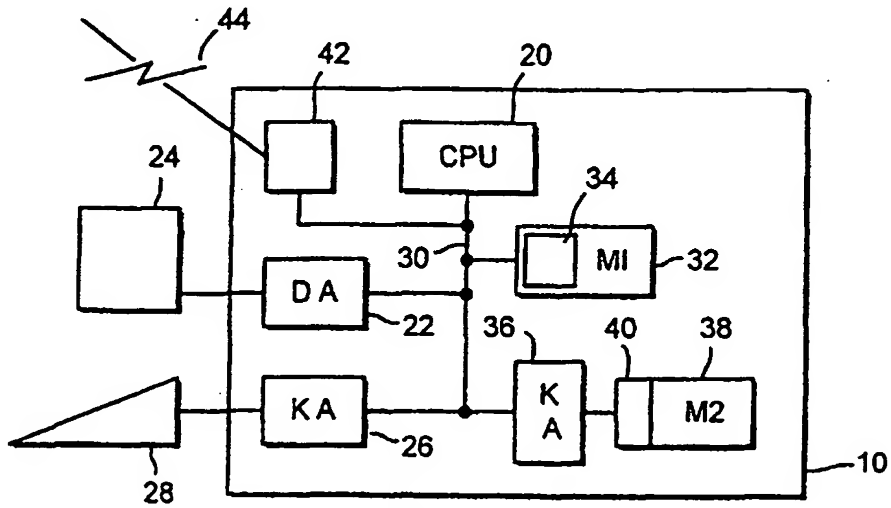

FIG. 1

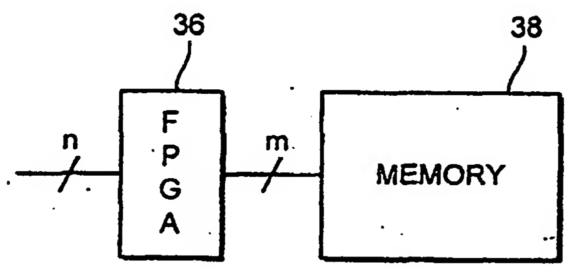

FIG. 2

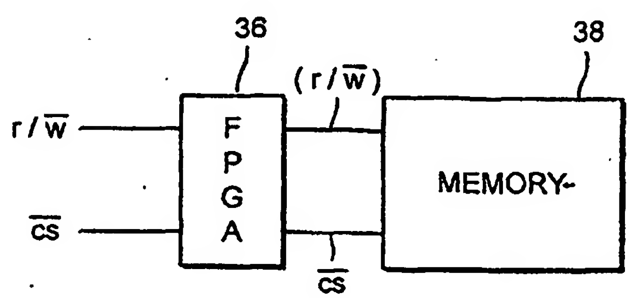

FIG. 3

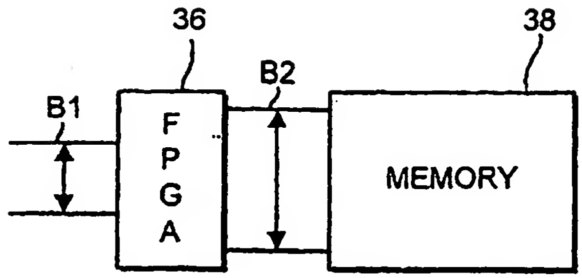

FIG. 4

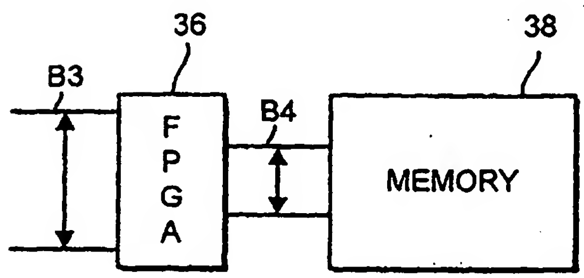

FIG. 5

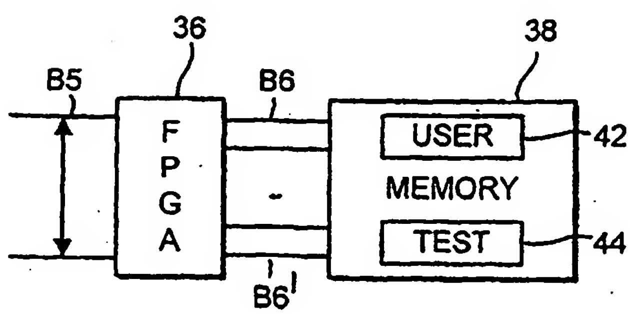

FIG. 6

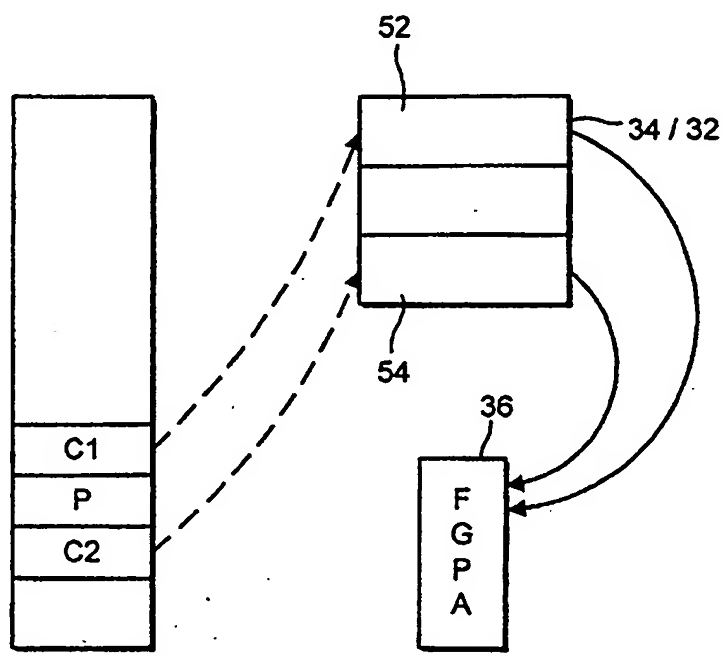

FIG. 7

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 851 358 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
01.07.1998 Bulletin 1998/27

(51) Int. Cl.^{6}: G06F 12/14

(21) Application number: 97310176.9

(22) Date of filing: 16.12.1997

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 31.12.1996 US 777256

(71) Applicant: Sun Microsystems Inc.
Palo Alto, California 94303-4900 (US)

(72) Inventor: Garnett, Paul Jeffrey
Merseyside, WA12 9PW (GB)

(74) Representative:
Harris, Ian Richard et al
D. Young & Co.,
21 New Fetter Lane
London EC4A 1DA (GB)

(54) Processing system security

(57) For controlling access to a system resource in a processing system, reprogrammable logic located between a bus and the resource is programmed in a first mode to permit access to the resource and is programmed in a second mode to at least restrict access to said resource via the bus. The resource can be a critical area of storage holding or identifying critical operational parameters or critical operational software relating to the processing system. The reprogrammable logic is preferably implemented using a field programmable gate array.

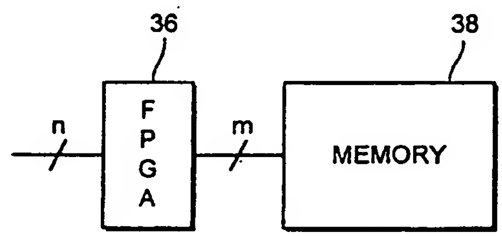

FIG. 2

EP 0 851 358 A2

Description

This invention relates to computer system security, and in particular to the protection of a critical resource within a processing system.

A computer system may be stopped from working by a software bug, or by a malicious act interfering with a critical system resource. The system resource can be a hardware resource, or can be software or data stored or otherwise retained within the computing system.

An object of the invention is to improve the security of a system by preventing, or making more difficult, access to a critical resource.

In accordance with a first aspect of the invention, there is provided a method of controlling access to a system resource in a processing system including a processing engine, said system resource and a bus, said method comprising steps of: in a first mode programming reprogrammable logic located between said bus and said resource to permit access by said processing engine via said bus to said resource; and in a second mode programming said reprogrammable logic to at least restrict access to said resource via said bus.

By reprogramming the reprogrammable logic to restrict access to the critical resource, accidental or deliberate corruption of the critical resource can be prevented or at least made more difficult, thus improving overall system security and integrity.

The resource can be a critical hardware component, although it will typically be an area of storage containing or identifying critical operational software or critical operational parameters of the system. The programming of the reprogrammable logic in the second mode can thus be used to block access selectively or totally to the storage area.

The storage area can comprise test software and/or test parameters relating to the processing system.

Preferably, the reprogrammable logic comprises a field programmable gate array. First programming information and second programming information can both be stored in memory in the processing system. Alternatively the first programming information is supplied externally to the processing system when required, and is not otherwise stored in the processing system. The latter alternative provides higher security and may be particularly useful for test purposes, whereby a test engineer could supply the first programming information in a test mode of operation.

In accordance with a further aspect of the invention, there is provided a processing system (for example a computer system) comprising a processing engine (for example a microprocessor or microcontroller), a system resource, a bus and reprogrammable logic located between said bus and said resource, said reprogrammable logic being programmable in a first mode to permit access by said processing engine via said bus to said resource and being programmable in a second mode to at least restrict access to said resource via said bus.

Particular embodiments of the invention will be described hereinafter with reference to the accompanying drawings in which like reference signs relate to like elements and in which:

Figure 1 is a schematic overview of a computing system;

Figure 2 is a schematic overview of one implementation of the present invention;

Figures 3-6 relate to further examples of implementations of the present invention;

Figure 7 is a schematic diagram illustrating the operation of the invention.

Figure 1 is a schematic overview of a computing system 10. The computing system comprises a processor 20 connected via a bus 30 to a number of resources. A display adaptor 22 enables a display 24 to be connected to the bus 30. Similarly, a keyboard adaptor 26 allows a keyboard 28 to be connected to the bus 30. A first memory M1 32 is connected directly to the bus 30. A further memory M2 38 containing a critical software resource 40 is connected to the bus 30 via reprogrammable logic, for example a reprogrammable gate array 36.

A communications adaptor 42 enables an external communications line 44 to be connected to the computer bus 30.

It will be appreciated that Figure 1 is merely a schematic overview of a computing system, and that an alternative computing system could have a different structure from that shown in Figure 1. In the example of Figure 1, and also in the examples set out hereinafter, the critical resource is assumed to be data, or software, stored in the memory M2 38. However, in other embodiments of the invention the critical resource could be a hardware component which is, for example, reconfigurable. An example of such a component could be a clock signal generator having a selectable clock rate. Alternatively, the hardware component could be a component of a system which is used only during certain modes of operation of the system. It may be desirable to prevent access to the hardware component by the user of the apparatus, permitting access only during a test or repair mode under the control of a service engineer.

In a preferred embodiment of the invention, the reprogrammable logic 36 is a reprogrammable gate array, such as a field programmable gate array (FPGA). In the following embodiments reference will be made to an FPGA 36, although other implementations of the device for restricting access to the resource may be provided.

Figure 7 illustrates an initial stage of operation of an embodiment of the invention such as that described in Figure 1. In particular, on initially booting the computing system, initialisation code 50 is operable to carry out the

initialisation of the system. At a point in the initialisation program, reference C1 is made to first code stored at locations 52 in the memory M1 32. This information is loaded from the memory 32 (which could, for example, be a read only memory) into the FPGA 36 to initially program the FPGA 36. The FPGA 36 is then responsive to signals provided over the computer bus 30 to provide a first mode for enabling access by programming code P to the critical resource 40 in the memory M2 to perform the appropriate initialisation functions. After execution of the code P, a second reference C2 to information stored at 54 in the memory 32 is made. This second information 54 is then loaded into the FPGA 36 overwriting the original programming of the FPGA 36. This reprogramming of the FPGA 36 can then prevent, or restrict further access to the critical resource 40 in a second mode.

Any one of many different models of commercially available reprogrammable can be employed as the FPGA 36 in an embodiment of the invention. The programming of the FPGA 36 will depend on the particular reprogrammable FPGA used and should be in accordance with the technical design specification for the FPGA concerned, as will be apparent to one skilled in the art.

Figure 2 is a schematic representation of one configuration of an FPGA 36 and a memory 38, the FPGA receiving an N bit bus, and being connected to the memory 38 by an M bit bus.

Figure 3 represents one specific embodiment of the invention with an FPGA receiving a read/write line r/w and a chip select line at cs and supplying the chip select and read/write lines to the memory 38. In this embodiment, the FPGA can be initially programmed to pass read and write signals, as well as the chip select signals to the memory 38. With the second programming (reprogramming) the FPGA can be arranged to pass only read signals, thus preventing writing to the memory 38. Alternatively, both read and write signals to the memory 38 could be prevented with the second programming. In either case system resource protection is provided either by preventing overwriting by the user or by completely blocking access to the memory 38.

Figures 4 and 5 indicate that the FPGA 36 can provide a translation between different sized buses. Thus, in Figure 4, a narrow bus B1 accesses the FPGA 36 but the FPGA 36 decodes the signals on the input bus B1 to provide individual decoded lines B2 for accessing the memory 38.

In Figure 5, a wide bus B3 accesses the FPGA 36, which only passes selected bits B4 of the address to the memory 38. In an embodiment of the invention, the FPGA provides a different translation in the first and second programming modes.

Figure 6 illustrates an arrangement where a bus B5 is provided at the input side of the FPGA 36. At the output side of FPGA, address lines represented at B6 and further address lines represented at B6' are provided. In an embodiment of the invention the address lines B6 provide address signals with both the first and second programming of the PFGA, whereas the address signals on lines B6' are either blocked or altered with either the first or second programming of the FPGA, whereby with the first programming, access to test software 44 within the memory 38 can be permitted, whereas with the second programming, access to the test programming is prevented and access is instead permitted to user software 42. It would then be possible, during initial design and testing, or during subsequent use for diagnostic purposes, for a test engineer to reprogram the FPGA with the original, first programming, and thereby permit access once more to the test software 44.

As illustrated schematically in Figure 1, the information 52, 54 is provided in the memory area 34 of memory 32. However, it may be that the information relating to the first programming of the FPGA may not be retained in the memory of the computer system 10 at all, but could instead be provided externally by a test engineer when testing of the system is required. In this way, it would not be possible for the user to seek out and possibly find the information required for programming the FPGA.

To provide additional security, it would be possible to encrypt the data stored in the areas 52/54 to make it more difficult for a user to find the information necessary for programming the FPGA 36.

An embodiment of the invention can provide security of operation in that accidental access to critical resources can be prevented during normal use. Also, an embodiment of the invention can make it significantly more difficult for the user to access the critical resources.

Although particular embodiments of the inventions have been described, it will be appreciated that the invention is not limited thereto, and many modifications and/or additions may be made within the scope of the invention.

Also, by way of further example, although in the embodiment described herein, the processing system is shown as a computer system, the processing system could be any apparatus or system having a computer-based, microprocessor-based or microcontroller-based control system.

Claims

1. A method of controlling access to a system resource in a processing system including a processing engine, said system resource and a bus, said method comprising steps of:

   in a first mode programming reprogrammable logic located between said bus and said resource to permit access by said processing engine via said bus to said resource; and
   in a second mode programming said reprogrammable logic to at least restrict access to

said resource via said bus.

2. A method according to Claim 1, wherein said resource is a critical area of storage, said second programming of said reprogrammable logic blocking access to said storage area.

3. A method according to Claim 2, wherein said storage area comprises critical operational parameters relating to said processing system.

4. A method according to Claim 2, wherein said storage area identifies or comprises critical operational software.

5. A method according to Claim 2, wherein said storage area comprises test software and/or test parameters relating to said processing system.

6. A method according to any preceding Claim, wherein said reprogrammable logic comprises a reprogrammable field programmable gate array.

7. A method according to Claim 6, wherein first programming information and second programming information are stored in memory in said processing system.

8. A method according to any preceding Claim, wherein said first programming information is supplied externally to said processing system when required, and is not otherwise stored in said processing system.

9. A processing system comprising a processing engine, a system resource, a bus and reprogrammable logic located between said bus and said resource, said reprogrammable logic being programmable in a first mode to permit access by said processing engine via said bus to said resource and being programmable in a second mode to at least restrict access to said resource via said bus.

10. A system according to Claim 9, wherein said critical resource is an area of storage, said reprogrammable logic, when programmed in said second mode, blocking access to said storage area.

11. A system according to Claim 10, wherein said storage area comprises critical operational parameters relating to said processing system.

12. A system according to Claim 10, wherein said storage area identifies or comprises critical operational software.

13. A system according to Claim 10, wherein said storage area comprises test software and/or test parameters relating to said processing system.

14. A system according to any one of Claims 9 to 13, wherein said reprogrammable logic comprises a reprogrammable field programmable gate array.

15. A system according to Claim 14, wherein first programming information and second programming information are stored in memory in said processing system.

16. A system according to any one of Claims 9 to 15, wherein said first programming information is supplied externally to said processing system when required, and is not otherwise stored in said processing system.

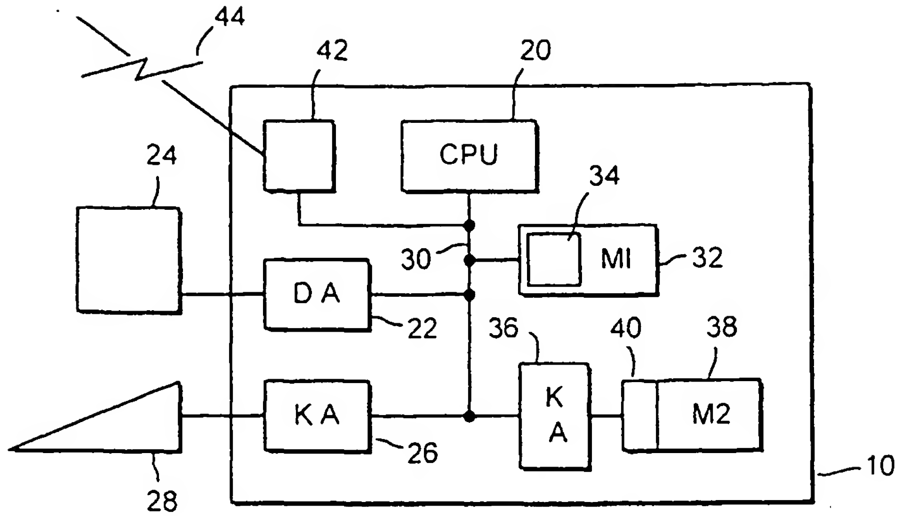

FIG. 1

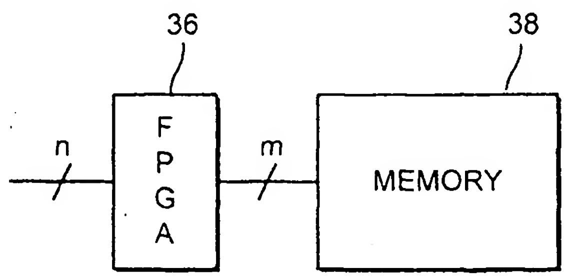

FIG. 2

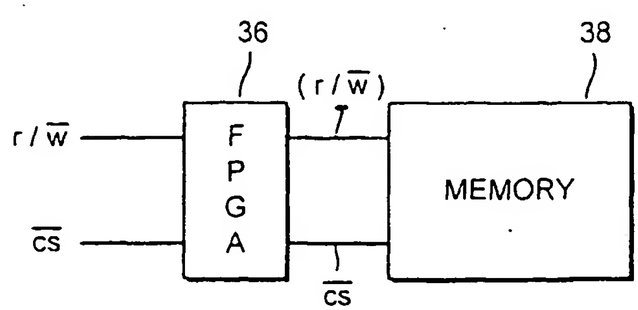

FIG. 3

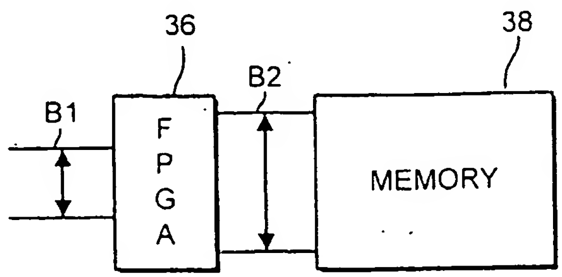

FIG. 4

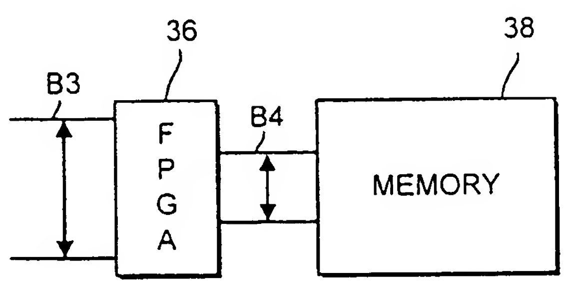

FIG. 5

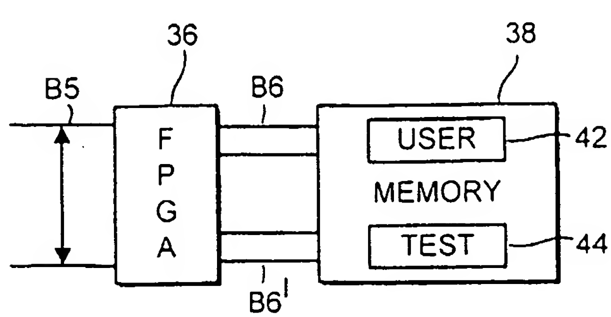

FIG. 6

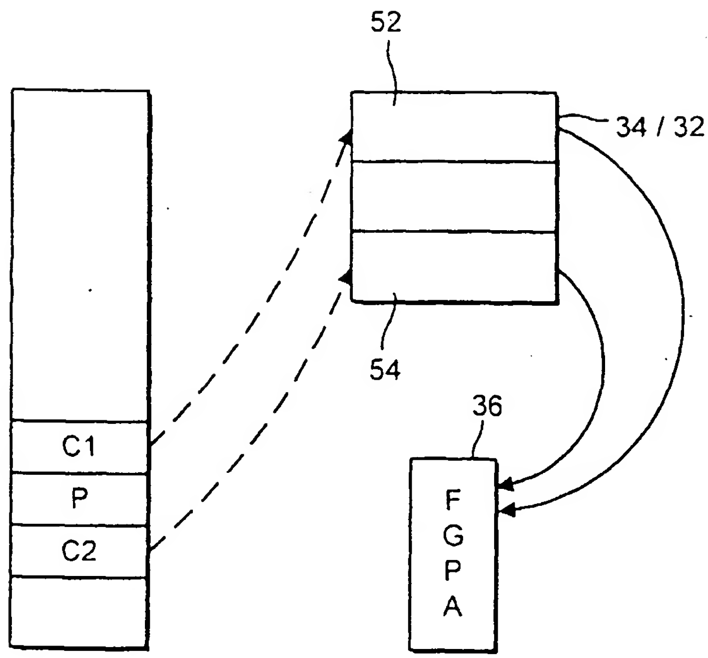

FIG. 7

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 851 358 A3

(12) EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
30.12.1998 Bulletin 1998/53

(51) Int. Cl.^{6}: G06F 12/14, G06F 1/00

(43) Date of publication A2:
01.07.1998 Bulletin 1998/27

(21) Application number: 97310176.9

(22) Date of filing: 16.12.1997

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 31.12.1996 US 777256

(71) Applicant:
Sun Microsystems, Inc.
Palo Alto, California 94303-4900 (US)

(72) Inventor: Garnett, Paul Jeffrey
Merseyside, WA12 9PW (GB)

(74) Representative:
Harris, Ian Richard et al
D. Young & Co.,
21 New Fetter Lane
London EC4A 1DA (GB)

(54) **Processing system security**

(57) For controlling access to a system resource in a processing system, reprogrammable logic located between a bus and the resource is programmed in a first mode to permit access to the resource and is programmed in a second mode to at least restrict access to said resource via the bus. The resource can be a critical area of storage holding or identifying critical operational parameters or critical operational software relating to the processing system. The reprogrammable logic is preferably implemented using a field programmable gate array.

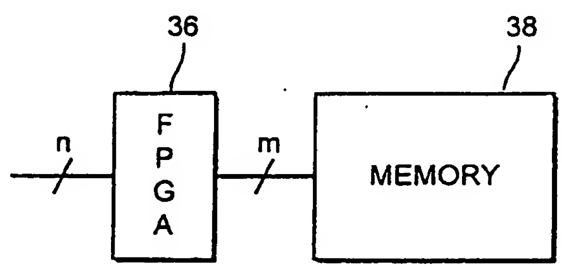

FIG. 2

EP 0 851 358 A3

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 97 31 0176

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
A	DE 27 33 531 A (SIEMENS AG) 1 February 1979 * page 2, line 24 - page 7, line 16; figure *	1-3,9-11	G06F12/14 G06F1/00
A	US 5 535 409 A (LARVOIRE ET AL.) 9 July 1996 * column 4, line 31 - column 7, line 33; figures 2-4 *	1-3,9-11	
A	EP 0 660 215 A (INTERNATIONAL BUSINESS MACHINES CORPORATION) 28 June 1995 * page 3, line 11 - page 4, line 28 * * page 5, line 1 - line 40 * * page 7, line 42 - page 9, line 28 * * page 25, line 30 - page 27, line 20; figures 1A,5,6 *	1,4,7,9, 12,15	

TECHNICAL FIELDS SEARCHED (Int.Cl.6)
G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
MUNICH	3 November 1998	Semple, M

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)